**Experiment Title. Write a program to recognize a valid arithmetic expression and to recognize the identifiers and operators present. Print them separately**

**Student Name:** Bhagath Reddy **UID:**18BCS696

**Branch:** BE-AI&ML-2  **Section/Group:**B

**Semester:**5  **Date of Performance:** 04-Oct-2020

**Subject Name:** System Programming Lab **Subject Code :**CSP-340

**1. Overview of Virtual Box:**

Virtual Box is very powerful cross-platform virtualization application.It installs on our existing Intel or AMS-based computers,whether they are running Windows,Mac OS X,Linux,or Oracle Solaris operating systems.It extends the capabilities of an existing computer so that it can run multiple threads,inside virtual machines at the same time.

Virtual Box is deceptively simple yet also very powerful.It can run everywhere from small embedded systems or desktop class machines all the way up to datacenter deployments and even Cloud environments.

**2. Tasks to be done:**

### Take the input from the c file

### Open file in the reading mode

### Fetch the arithmetic operators

### Fetch the identifiers

### Display the same.

**3. Steps for practical**: (Mention the steps for each and every task)

S1)Install a Flex tool.(here I have used Ubuntu LTS(20.04) environment for Flex)

S2)If using Ubuntu,install flex on it by using the command -sudo apt install flex bison

S3)Type cd c/mnt command

S4)Then go to that folder you have saved your lex file.

S5)Make a lex file using command -vim <filename>.l

S6)Write the Lex program there by following a proper format.

%{ Definitions %}

%%

{ Rules }

%%

{ User Subroutines }

**Lex Program**

%{

#include<stdio.h>

int arithCount=0;

int identiCount=0;

int keywordCount=0;

%}

arithmetic [+-/\*%=]|(\+\+)\*|(\-\-)\*

identifiers [[:alpha:]\_]\*

keywords double|if|else|for|do|while|int|char|float|long|stdio|main|return|#include<stdio.h>

delemiter [,\(\){::};]

print printf\((.)\*\)

%%

{delemiter} {}

{keywords} {

printf("\b%s is a keyword\n",yytext);

keywordCount++;

}

{arithmetic} {

printf("\b%s is an arithmetic operator\n",yytext);

arithCount++;

}

{identifiers} {

printf("%s is an identifier\n",yytext);

identiCount++;

}

{print} {}

%%

int yywrap()

{};

int main()

{

yyin=fopen("file.c","r");

yylex();

printf("\nThere are %d arithemetic operators in total",arithCount);

printf("\nThere are %d identifiers in total",identiCount);

return 0;

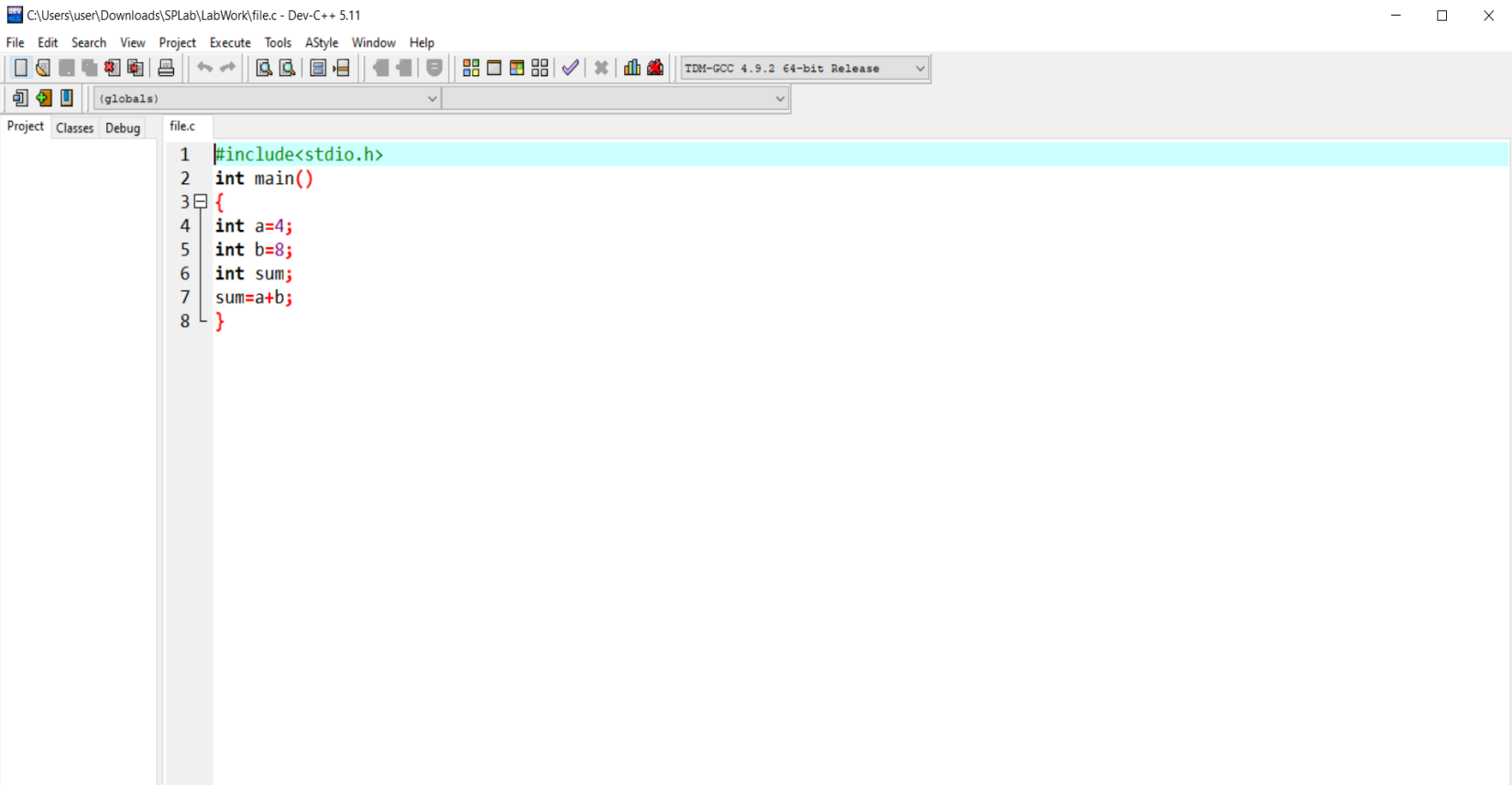
}

**4. Screenshots:**

Program

****

C program:

****

Output:

****

**5. Commands used:**

Commands used on Ubuntu to run the LEX program:

Create your file using vim command vim <file\_name>

1. **vim prog.l**

This command will create the file of name ‘exp3.l’

1. **lex prog.l**

To give the exp3.l file as an input.

1. **gcc lex.yy.c**

Used to run .c file

**4)./a.out**

Command used for lexical analyzer

**6. Result/Output/Writing Summary:**

In this experiment the following tasks got completed:

a) Taking input from c file

b) Opening it in read mode

c) Compilation of .c file using lex tool.

d) Difference between identifiers and keywords got established.

At the end of this experiment we have learnt about the REGEx pattern of identifying Identifiers and arithmetic operators from a given .c file using lexical analysis.

**Learning outcomes (What I have learnt):**

**1 .**Learnt about the REGEX patterns and their workings.

**2.** Learnt about the REGEx pattern of matching and identifying Identifiers.

**3.** Learnt about the REGEx pattern of matching and identifying arithmetic operators.

**4.** We learnt about the compilation of .C file with the help of lex command i.e gcc lex.yy.c

**5.** We also have learnt about the crude and elementary idea of delimiters and commas used while writing a program.

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |
|  |  |  |  |